SRI International

Applying an AI Planner to Military Operations Planning

Technical Note No. 534

January 12, 1993

By: David E. Wilkins, Sr. Computer Scientist
Artificial Intelligence Center
Computing and Engineering Sciences Division

Roberto V. Desimone, Sr. Computer Scientist
Information and Telecommunications Sciences Center
Information, Telecommunications & Automation Division

		Form Approved
Report Documentation Page		OMB No. 0704-0188

1. REPORT DATE **12 JAN 1993**	2. REPORT TYPE	3. DATES COVERED **00-01-1993 to 00-01-1993**

4. TITLE AND SUBTITLE **Applying an AI Planner to Military Operations Planning**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SRI International,333 Ravenswood Avenue,Menlo Park,CA,94025**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**34**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Abstract

This paper describes a prototype system for quickly developing joint military courses of action. The system, SOCAP (System for Operations Crisis Action Planning), combines a newly extended version of an AI planning system, SIPE-2 (System for Interactive Planning and Execution), with a color map display and applies this technology to military operations planning. This paper describes the Socap problem domain, how SIPE-2 was used to address this problem, and the strengths and weaknesses of our approach.

Contents

List of Figures

1 Introduction

This paper describes the cooperative work taking place under two projects in the DARPA/RL Planning Initiative to produce a prototype system for quickly developing joint military courses of action. One project, at the Artificial Intelligence Center of SRI International (SRI), is developing a novel planning and execution system based on several high-performance artificial intelligence (AI) technologies. Another project, at SRI's Information and Telecommunications Sciences Center, is applying these evolving technologies to planning problems of the U.S. Central Command (CENTCOM).

As a result of this collaboration, SRI developed SOCAP (System for Operations Crisis Action Planning). It combines a newly extended version of an AI planning system, SIPE-2 (System for Interactive Planning and Execution), with a color map display and applies this technology to military operations planning. SOCAP was connected to the Dynamic Analysis and Replanning Tool (DART) via FMERG[1] to constitute the second Integrated Feasibility Demonstration (IFD-2) for the Planning Initiative. FMERG elaborates in more detail the Socap transportation plans, and then uses DART to run a simulation to determine their transportation feasibility. IFD-2 was demonstrated in early 1992 both at CENTCOM and at the Pentagon. The aim was to demonstrate the feasibility of applying the Sipe technology for the generation of large-scale military operations plans (OPLANs).

The objective of the Socap research effort is to develop decision aids to enable military planners to produce more flexible and accurate joint military courses of action in less time when responding to a crisis. To develop prototypes that can have a provable impact in an operational environment, it is necessary for them to be tested by the military community. Thus, SOCAP was developed by consulting with military operation planners at CENTCOM to elicit their requirements and their knowledge of the planning

---

[1]FMERG (Force Module Expanded Requirements Generator) was developed by BBN and ISX under the DARPA/RL Planning Initiative to bridge the gap between the description of major forces in a course of action and the description of their corresponding TPFDD-level components required for DART, which is in operational use and tests a TPFDD (see Section 2) on a transportation feasibility model.

process. In addition, information from other sources and publically available texts were used to develop this application. SOCAP successfully generated employment plans for dealing with specific enemy COAs, and expanded deployment plans for getting the relevant combat forces, supporting forces, and their equipment and supplies to their destinations in time for the successful completion of their mission. Input to the system includes threat assessments, terrain analysis, apportioned forces, transport capabilities, planning goals, key assumptions, and operational constraints.

This paper describes the Socap problem domain, how SIPE-2 was used to address this problem, and the strengths and weaknesses of our approach. The cooperative aspect of the work is important: the domain requirements of the application are driving some of the research in developing AI planning technologies, and these evolving AI tools and techniques are being transferred to the application as rapidly as possible.

2 Military Operations Crisis Action Planning

This section briefly describes the Socap problem domain. The military must manage crises. Good crisis management is characterized by quick response, decisive action, and flexibility to adapt to the changing situation. Developing a good course of action (COA), and modifying it as necessary, must take into account a number of factors: approaches used in past cases that have worked well, novel features of the new situation, differing priorities for subparts of the crisis, and feasibility of suggested COAs. A COA should describe an *employment* plan for dealing with one or more enemy COAs, and should identify a *deployment* plan for moving the relevant combat forces, supporting forces, and their equipment and supplies to their destinations in time for the successful completion of their mission.

Currently, the military crisis action planning process involves several phases. When a crisis occurs that requires a military option to be considered, the crisis is assessed and the situation reviewed to determine if military action is required. If it is, then tentative COAs are generated based on doctrine, past exercises, and existing concept and operations plans. Various estimates are developed for personnel (J1), intelligence

2

(J2), operations (J3), logistics (J4), and command and control (J6). The estimates and recommendations are presented for approval, and the commanders' estimate is constructed from these. Final approval is required before a complete OPLAN is generated.

Next, the force composition, logistics to support the mission, and all the transportation needs are defined, planned, synthesized, and simulated. Time-phased force deployment lists are generated based on force module libraries or previously developed deployment lists. These are routed for a detailed transportation feasibility analysis. The refined TPFDD (Time-phased Force Deployment Data) is approved for both operational and transportation feasibility, and operation orders are developed. Finally, the operation orders are executed, and monitored.

The crisis action planning process is a distributed, interactive process. Accurate, timely, and secure exchange of information among geographically separated commanders is required to produce the best plans in the shortest time. Also, the planning process, as defined above, provides for the maximum reuse of plans where possible. Uncertainty is inherent in planning a response to a developing crisis and must be handled adequately if plans are to be robust. Other requirements for automating the joint military operations planning process are given elsewhere [2].

A typical OPLAN contains approximately a few hundred actions, describing the employment and deployment of force modules chosen to deter or counter the enemy threat. A typical TPFDD contains a few thousand entries, each describing which unit, or part of a force module, is being deployed, where and when it arrives at its destinations, and by which means it is transported. Each enemy threat may be deterred or countered with a wide variety of operations that apply differing levels of aggression. These operations can vary from a show of force, to a blockade or quarantine, to a complete defensive or offensive operation. There is also a wide variety of units that have differing capabilities suitable for many different operations.

The number of possible COAs is enormous: developing a COA involves choosing operations at many levels of detail, military units and resources, and locations and times of these operations. In addition, rules of engagement need to be observed, oper-

3

ational constraints (e.g., troop limits) need to be satisfied, permissions for overflight of Allied airspace must be observed, and key assumptions need to be made explicit, such as the nonintervention of third-party forces. Most of these conditions are provided either in the mission statement or as planning guidance.

For demonstration purposes, SOCAP encodes a typical scenario that involves the use of U.S. forces to protect the territorial integrity of a friendly country from a neighboring enemy. A joint force is chosen that has significant defensive, rather than offensive Capability. The mission statement normally identifies a D-day by which date the ground forces should be deployed, and a further date by which the defensive operations should have been completed. Associated with each action is an estimate of its duration, estimates of the appropriate size of unit it requires, and the location and region where it takes place. This information is required to "source" the military units in the plan (i.e., identify actual units); later, this information helps to determine the effects of deployment delays on subsequent actions.

The primary role of the ground forces is to provide a defensive screen near the border with the enemy, and to secure key terrain in this border region. The role of the Navy is to secure sea lines of communication between the friendly territory and its trading neighbors, and to protect major seaports. The role of the Air Force is to provide air defense over the territory. In addition, the Navy and Air Force have subsidiary roles to protect the deployment of ground forces and to support some of their operations, as with amphibious landings and close air support. Such interdependencies between actions make it possible to predict the impact of various changes in the situation or the reduced availability of specific units.

3 The SIPE-2 Planning System

SRI's AI Center has been conducting research into planning and problem-solving systems for the past three decades. SIPE–2 (System for Interactive Planning and Execution)[2] is the most advanced of SRI's planning systems, and provides the core

---

[2]SIPE-2 is a trademark of SRI International.

reasoning engine for plan generation in SOCAP. This section briefly describes the features of SIPE–2 that are most important to SOCAP; the system has been described in detail elsewhere [5, 6, 8].

SIPE–2 provides a formalism for describing actions and utilizes the knowledge encoded in this formalism, together with its heuristics for handling the combinatorics of the problem, to generate plans for achieving given goals. Given an arbitrary initial situation, the system either automatically or under interactive control generates plans, possibly containing conditionals, to achieve the prescribed goals. The generated plans include causal information so that during plan execution the system can accept descriptions of arbitrary unexpected occurrences and modify its plans to take these into account. Unlike expert systems, SIPE–2 is capable of generating a novel sequence of actions that responds precisely to the situation at hand.

SIPE–2 is a nonlinear AI planning system that plans at different levels of abstraction. Because this technology is generic and domain-independent, it has the potential to affect a large variety of problems in fields as diverse as manufacturing, construction, and the military. Unlike most AI planning research, heuristic adequacy (efficiency) has been one of the primary goals in the design of SIPE–2. This has enabled its application to many domains, including the blocks world, planning the actions of a mobile robot, planning the movement of aircraft on a carrier deck, travel planning, construction tasks, the problem of producing products from raw materials on process lines under production and resource constraints, and the military operations planning described in this paper.

SIPE–2 has implemented several extensions of previous planning systems, including the use of constraints for the partial description of objects, the incorporation of heuristics for reasoning about resources, and replanning techniques. Its formalism allows description of planning and simple scheduling problems in terms of the goals to be attained and the various activities that can be undertaken to achieve these goals. One of the most powerful heuristics for avoiding combinatorics in SIPE–2 is the ability to avoid frequent consistency checks by temporarily producing invalid plans. The system relies on *plan critics* that check for and correct problems in these plans at certain intervals.

5

state of real world — actions one might take in real world — problems in real world

Predicates, State Rules, Sort Hierarchy

Operators, Causal Rules

Planner

Problems

sequence of actions
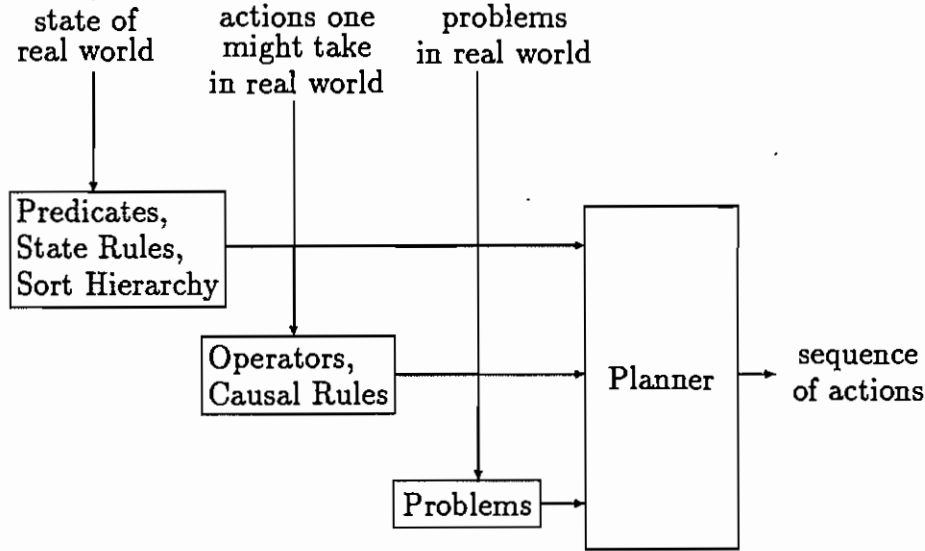
Figure 1: SIPE-2's View of the Planning Problem

3.1 Describing the Domain in SIPE-2

The inputs and outputs of SIPE–2 are depicted in Figure 1. While the inputs to the planner attempt to model the "real world," current AI techniques cannot handle the full complexity of our everyday world and generally it is an abstraction of the real world that is represented. The vertical arrows in the figure indicate the relationship of representation, by which entities in the world are encoded internally in the planning system. The output is a plan — a partially ordered set of *primitive* actions and conditions to be carried out. Conditions the plan expects to be true at certain times are also included in the plan. A plan can be represented as a directed, acyclic graph in which each node is an action or condition, and each edge represents a temporal ordering.

SIPE–2 takes as input a description of the initial state, a description of a set of actions, a problem descriptor, and a set of rules describing the domain. The initial state consists of a *sort hierarchy*, a *world model*, and a set of deductive rules. The sort hierarchy represents invariant properties of perpetual objects, describes the classes

6
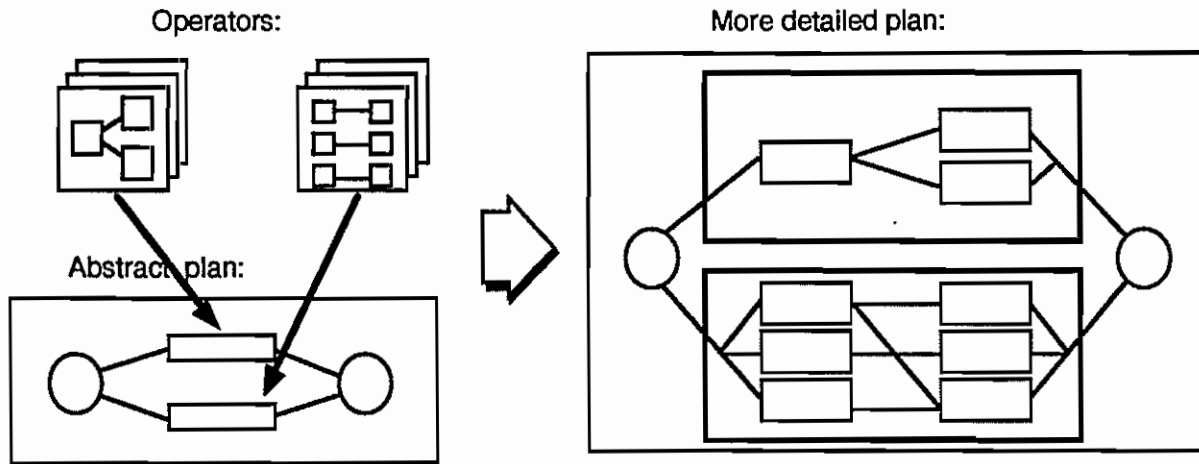
Figure 2: Using Operators to Expand a Plan

to which an object belongs, and allows for inheritance of properties. In SOCAP the sort hierarchy is used to encode information such as terrain analysis, the attributes of the combat forces available, and transport capabilities. The world model is a set of predicates which hold over objects in the sort hierarchy. Some predicates are given explicitly in the data base, and others are deduced by applying deductive rules to the data base. Predicates encode information such as operational constraints, assumptions, and intelligence reports.

An *operator* is the planner's representation of actions or abstractions of actions that may be performed in the domain. In SOCAP, operators vary from abstract strategies to specific military operations that can achieve employment or deployment goals. An operator is input in the system's formalism and specifies the conditions under which an action is appropriate, the constraints on the objects in the action, a set of instructions for performing the action, and the main effects accomplished by the action (additional context-dependent effects are deduced by the system from deductive rules).

To produce a plan, the planner instantiates operators by binding their variables, combines these instantiations by ordering them, and then adds additional constraints to avoid problematic interactions between actions. For example, if a potential resource conflict is detected between two actions, the planner may order the actions to execute sequentially or may constrain the action to use different resources. The left side of

Figure 2 depicts SIPE–2 choosing one of a set of operators for each goal in an abstract plan. This produces the more detailed plan on the right to which the planner has added an ordering link to avoid a problem it has detected (e.g. a resource conflict). The graphs in the figure represent plans where each rectangle is a goal or action, and the edges are temporal ordering links running from left to right. Section 5 describes a Socap operator and its application during plan generation.

3.2 The User Interface in SIPE-2

Because of the size and complexity of the Socap problem domain and the plans produced, the user interface is critically important. It proved necessary to have two different interfaces — one for system engineers developing the implementation (the Sipe interface), and another for the military users (the Socap interface). The map-based Socap interface is described in Section 5; the Sipe graphical user interface (GUI) is described here.

Several new interface capabilities were needed to cope with the size of the military planning problem. For example, the sort hierarchy needed to be displayed graphically as a tree to help ensure its correctness. The world model, operators, and objects also needed to be displayed graphically. With commands to aid in generating plans, viewing complex plans and other information as graphs on the screen, and following and controlling the planning process, it was necessary to provide easy access to all the commands. To support this, we carried out a complete redesign and reimplementation of the Sipe GUI as part of this research. The new interface is built on Grasper-CL[3] and uses a set of noun menus, each of which brings up a verb menu. The new interface provides many new capabilities and makes them more easily accessible through the noun/verb menus. Figure 3 shows the functionality of the new Sipe interface by

---

[3]Grasper-CL is a trademark of SRI International. Grasper-CL is a programming-language extension to Lisp that introduces graphs — arbitrarily connected networks — as a primitive data type. It includes procedures for graph construction, modification, and queries as well as a menu-driven, interactive display package that allows graphs to be constructed, modified, and viewed through direct pictorial manipulation.

PROFILE	PROFILE	PROFILE	PROFILE	PROFILE
PROFILE	DOMAIN	DOMAIN	DOMAIN	DOMAIN
DOMAIN	**DOMAIN**	PLAN	PLAN	PLAN
PLAN	PLAN	**PLAN**	PLAN	PLAN
DRAWINGS	DRAWINGS	DRAWINGS	**DRAWINGS**	DRAWINGS
NODE	NODE	NODE	NODE	**NODE**
	INPUT			
TRACE	RESET	SOLVE:	DRAW:	FIND
		automatic	operator	PRINT
PLANNING	MODIFY	interactive	problem	
EFFICIENCY	FIND	continue	plan	PREDECESSORS
NUMERICAL	CLEAR plans		world	SUCCESSORS
		EXECUTE	objects	PARALLEL
PRINTING	LIST:			
	operators	ABSTRACT	SELECT	RESOURCE
DRAWING	problems	REGROUP	DESTROY	ARGUMENT
SIZE NODES	plans			PREDICATE
	nodes	PRINT	NEW VIEW	
	worlds	RENAME	RENAME	RESHAPE
	objects	DESTROY	REDRAW	MOVE
	contexts			ALIGN
		-> ACT	BACKUP	
	PRINT:		REVERT	
	operator		RESCALE	
	problem		HARDCOPY	
	plan			
	node		GRAPH:	
	world		select	
	object		create	
	context		destroy	
			input	
	-> ACT		output	

Figure 3: SIPE-2 GUI Noun and Verb Menus

depicting the various menus of the GUI (the five nouns are at the top of each menu, and the verbs for the selected noun are below).

The five nouns let the user choose the level at which the verbs (commands) will operate. The PROFILE noun activates commands for setting defaults that allow the user to customize the behavior of the planner. The DOMAIN noun activates commands that apply to the problem domain as a whole, e.g., inputing and inspecting a domain or solving a problem. The PLAN noun activates commands that apply to a specific plan, including executing a plan and solving a problem to produce a plan. The DRAWINGS noun activates commands that draw various data structures. Objects that can be drawn include plans, operators, the sort hierarchy, the initial world model, and problems. Finally, the NODE noun activates commands that apply to specific nodes in the currently drawn plan. Each action and goal in a plan is represented as a node in the graphical drawing of the plan.

The commands in the NODE menu are particularly helpful when analyzing large plans which can be difficult to interpret. These commands are used to find and modify

9

Figure 4: Socap Plan Highlighting Resource: Fennario Port

nodes, to find resources and predicates at nodes, and to view, by highlighting nodes, various properties of a plan. The modification options are useful for customizing the appearance of the drawing on the screen — the user can move nodes around with the mouse to make the drawing look exactly as desired. An example of using highlighting of nodes effectively is the *Resource* command which can be used to highlight all the nodes that mention Fennario Port, effectively showing the schedule for that port (as in Figure 4). While only a part of the plan is shown in the figure, the GUI provides powerful capabilities for panning over the entire plan, locating a particular part of the plan, or looking at a birds-eye view of the entire plan.

3.3 Interactive Planning

While SIPE–2 can generate plans autonomously, it also permits the user to interact with and control the planning process. In military operations planning, this feature is critical since the plans are large and complex, and there are many experienced human planners whose expertise can be used. Because of this, a new and more powerful interactive search algorithm was designed and implemented. In particular, the interactive search is now tightly integrated with the GUI.

The GUI allows the user to control, when desired, many aspects of the planning process during joint military operations planning. The user can decide when to apply certain planning algorithms (known as *plan critics*), can choose which operator to apply after the system has determined which ones are applicable, and can choose which object (e.g., a military unit or location) to use for a particular action. At any of these choice points, the complete power of the GUI is available for inspecting data structures (e.g., an operator can be drawn before choosing it). The option of planning automatically for either one abstraction level or for the rest of the plan is available. All choices are kept so that different alternative plans can be developed simultaneously. The ability to understand what SIPE–2 is doing is enhanced by the ability to highlight a node on the screen whenever the system is making a decision about that node, e.g., planning for the node, choosing an operator for the node, or choosing instantiations for the variables at that node.

The combination of these features allows the user to understand and effectively control the planning process in domains as large as military operations planning.

3.4 Scheduling

Robust solutions to complex, real-world problems require both planning and scheduling. In this section, we describe how SIPE–2 differs from most scheduling systems and why it is necessary to combine planning and scheduling. If the set of actions that must be performed, i.e., the process network, is known beforehand then it is a scheduling problem to assign resources and times to these actions. On the other hand, if the set

of actions must be generated based on the current situation and the current goals, then the problem solution also involves planning which requires that the system reason about how the world changes as scheduled events occur. This is the situation in military operations planning.

Other features of a scheduling problem can require planning; in fact, problems are often not separable into distinct planning and scheduling problems. For example, if some subset of the constraints in a scheduling problem will change depending on how another subset is satisfied, e.g., if the constraints on the afternoon's schedule will change depending on how the constraints on the morning's schedule were satisfied, then such a scheduling problem becomes a planning problem. Similarly, if a system is to modify an existing plan or schedule in response to unexpected occurrences during execution, then reasoning about the effects of actions and the causal relationships between actions becomes necessary to determine which subplans remain valid or are amenable to modification. In most real-world situations, unexpected occurrences are the norm during operations, and it is important to modify plans quickly in response to these occurrences.

In SIPE-2, time is treated as a consumable resource that can be consumed but not produced, and whose consumption over parallel tasks is nonadditive [5, 6]. The numerical value can be a list of numbers customized for the problem. For example, time can be represented as (days hours minutes seconds). Each action specification may have *start-time* and *duration* slots containing variables with numerical constraints on them that are satisfied by the planner. In particular, a variable can be constrained to be the value computed by a Lisp function which is generally how durations are computed.

SIPE-2 will calculate specific values for time variables only when the constraints force a particular value; otherwise, the allowable range is computed. When several such variables have ranges, it is a scheduling problem to determine optimal or desirable times for each variable. The planner does not have built in algorithms for doing this — good algorithms often require tailoring to a specific domain. SIPE-2 does have hooks built in so that external scheduling modules can assign these times. We are currently working

12

on developing such a tie with the scheduling systems being developed at Carnegie-Mellon University (CMU) [4]. However, many research issues must be addressed to maximize the benefit from such an integration (see Section 6).

SIPE-2 has several techniques for establishing relative orderings of actions. These techniques include inserting ordering links to avoid resource conflicts, using one action to meet several different requirements by ordering them after the action, and coordinating separate subplans by adding ordering links to goals that have been declared as *external*. SOCAP makes heavy use of the latter two techniques; in fact, SIPE-2 was extended to do more sophisticated reasoning about when and how to achieve several different requirements by one action in order to support SOCAP.

From the scheduling viewpoint, SIPE-2 creates the project network that must be scheduled. In addition, it does the part of the scheduling that can be determined from its hard constraints, and also uses heuristics for adding relative ordering links. Depending on the specific circumstances, this may do much of the work of establishing a schedule (as in SOCAP), or it may leave considerable work to be done by another scheduling module. As described in Section 6, we would ideally like to have a scheduler interact with the planner during planning, providing values for its time variables based on scheduling algorithms.

4 SOCAP – System for Operations Crisis Action Planning

The objective of the IFD-2 version of SOCAP was to demonstrate the feasibility of applying the Sipe technology to the generation of large-scale military OPLANs. This was accomplished by developing decision aids for the generation of more flexible and accurate joint military COAs. To date, no research or development activity has integrated a generative AI planning system into an operational environment. While SOCAP stresses the demonstration of generative planning techniques, subsequent phases of this research will emphasize replanning techniques.
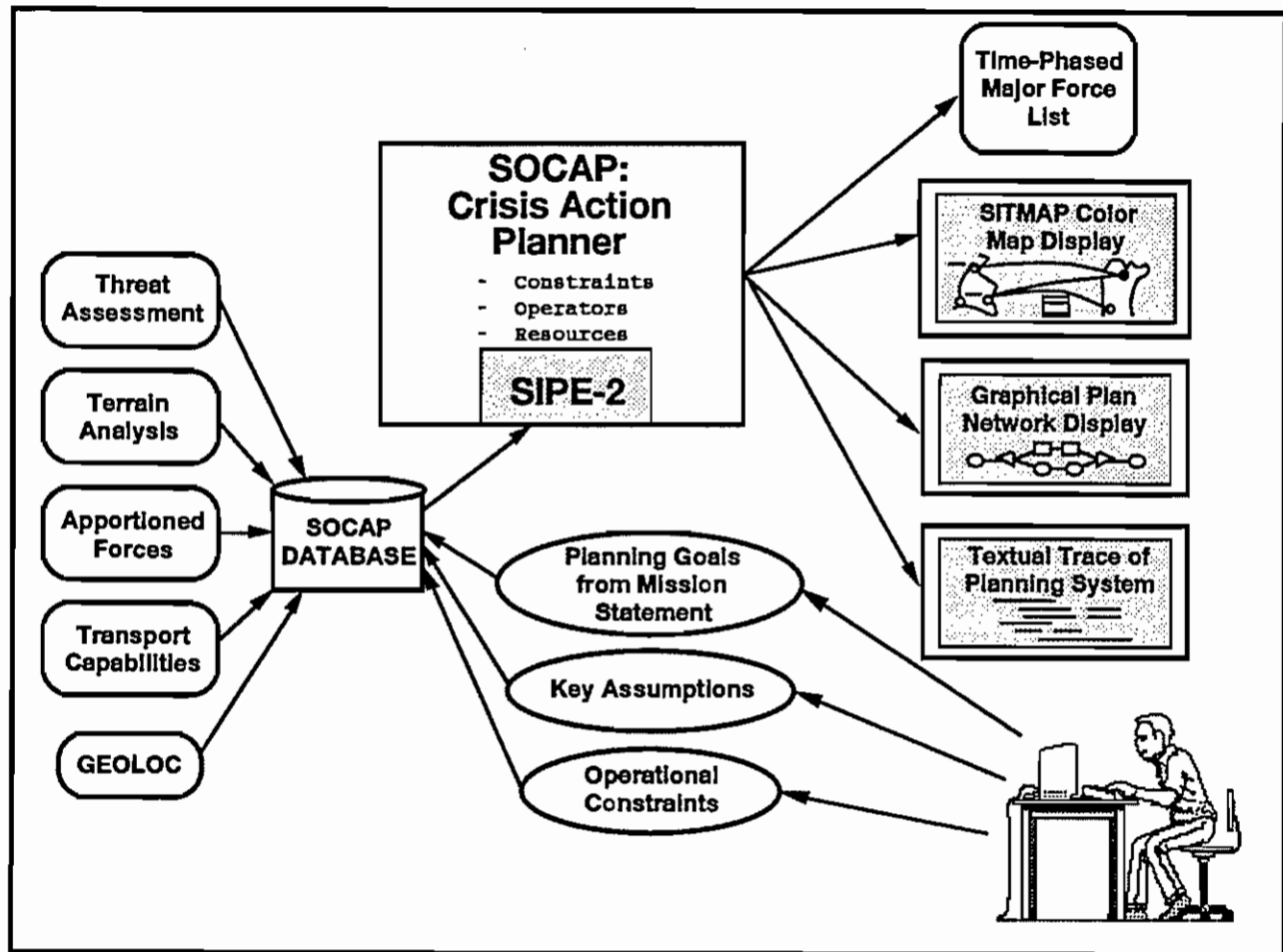
Figure 5: SOCAP Architecture

SOCAP includes an application of SIPE-2 to military operations planning together with a user interface tailored to this domain using a situation-map display system. Figure 5 shows the architecture, highlighting the necessary inputs for the generation of COAs and OPLANs, the available outputs, and the user interaction. The following inputs should be fed into the Socap database from available military databases:

Threat assessment: list of enemy threats, locations, and dates.

Terrain analysis: information on terrain features affecting mobility and observability.

Apportioned forces: list of combat forces available for planning purposes.

Transport capabilities: list of available assets.

Other inputs would come from the mission commander or his/her joint staff:

Planning goals: list of goals that match mission statement.

Key assumptions: e.g., rules of engagement, non-intervention of third party forces.

Operational constraints: e.g., overflight privileges, troop limits in country.

Most of the above information is inherently dynamic, and is best represented in SIPE-2 as first-order statements, such as (threat-enemy 1stArmBde route-A 25), which states that there is an enemy threat unit, the 1st Armored Brigade, whose avenue of approach is denoted by route-A, and that the enemy unit should be countered by Day 25. There are over 1200 predicate statements in the Socap knowledge base. However, a great deal of the available data are static (i.e., they do not change over time as actions are executed), and for efficiency reasons are best represented in the sort hierarchy and not as predicate statements.

The major (parent) classes represented in the static database are:

Forces: including combat, combat support, and combat service support.

Transports: including air, sea, and ground.

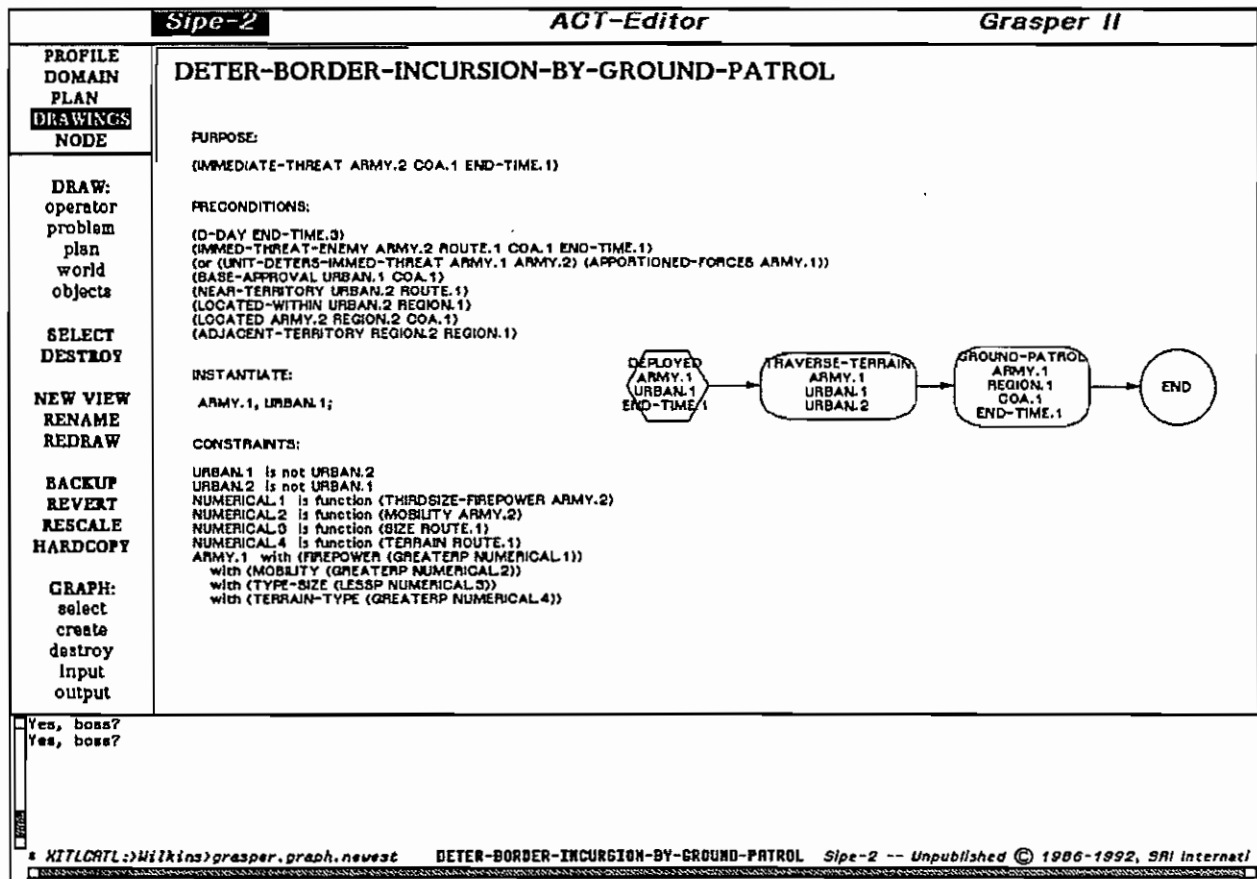**Territory:** including land, sea, and air.

Figure 6: Deter Border Incursion Operator in SIPE-2 Interface

All other entities, such as cargo, equipment, aircraft, ships, routes, regions, urban areas, etc., are either subclasses derived from the above three classes or are properties of classes. For example, cargo requirements and combat capabilities for specific combat forces should be denoted as properties of these forces. There are over 200 classes and 500 objects in the IFD-2 knowledge base. For efficiency reasons, it is important to represent static information using the sort hierarchy whenever possible. This is discussed in Section 5.

SOCAP requires a large set of operators to describe military operations that can achieve specific employment or deployment goals. For instance, there are a variety of military operations for deterring an enemy army, navy, or air force. Each of these operations may be represented by a different Sipe operator which all have the common

16

effect of deterring an enemy force. However, they may have different sets of preconditions or different resource requirements that need to be satisfied before they can be used in a plan.

There are operators at every level of abstraction. The ability of SIPE-2 to plan at different levels of abstraction naturally maps onto the following levels of the operations planning process:

Level 1: Select mission type.

Level 2: Identify threats and their locations.

Level 3: Select employment operations, major forces, and deployment destinations.

Level 4: Add deployment actions.

Section 5 describes how important the hierarchical nature of the planning process is to the success of SOCAP.

4.1 Applying Operators

It is the operators that capture most of the domain-specific planning capability. There are currently around 50 Sipe operators in the Socap knowledge base.[4] Each operator brings into the plan an appropriate network of military actions and subgoals that together achieve the purpose of the operator. We have made use of these subgoals to highlight dependencies on other parts of the plan to provide supporting military actions such as resupply, close air support, or artillery support.

Figure 6 shows a typical operator for deterring an enemy ground threat as drawn by the GUI. Here we briefly describe how this ground-patrol operator is used to generate a more detailed plan by the process depicted in Figure 2. Figure 7 shows an abstract plan produced after the second level of planning during which enemy threats are detected.

---

[4]We have recently added 20 deductive operators for correctly deducing location changes.
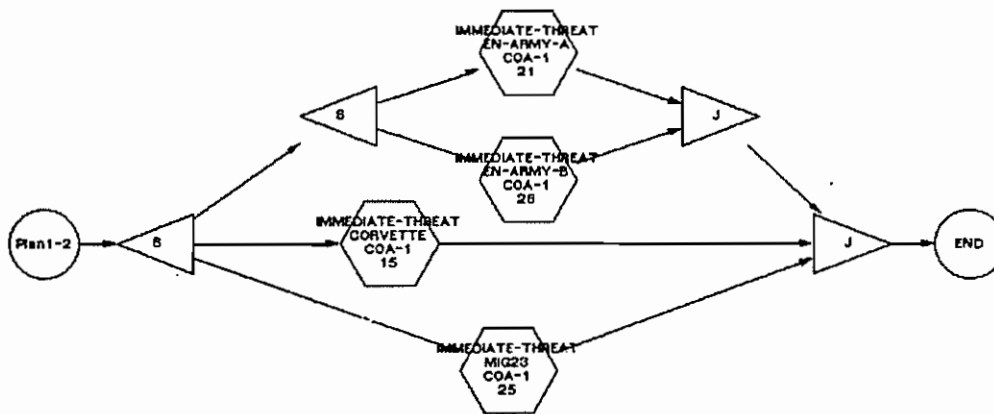
Figure 7: Plan after Threat Identification (Level 2)

This plan is a subset of the plan developed for IFD-2, and has four parallel goals of deterring each of four immediate enemy threats. The second goal, (immediate-threat enemy-army-B coa-1 26), requires the enemy army to be deterred by Day 26. The operator shown in Figure 6 is one of four operators applicable to this goal because its *purpose* matches the goal. The planner chooses this ground-patrol operator to expand the plan to level 3, and binds enemy-army-B to the army.2 variable in the operator. The expansion succeeds because the preconditon of the operator is true; matching the precondition constrains variables, e.g. urban.2 is constrained to be near route.1 and within region.1. Additonal constraints (shown in Figure 6) are posted by application of this operator, e.g., army.1 must have appropriate firepower and mobility.

The graph on the right side of Figure 6 depicts the instructions for achieving deterrence at the next level of detail, which consists of the subgoal (a hexagonal node) of deploying army.1, followed by the action (a capsule node) of traversing from urban.1 to urban.2, followed by the action of doing ground patrols in region.1. Figure 8 shows the plan at the next level when all threats have been countered. Both the first and second threat have been countered by the ground-patrol operator, while the third and fourth threats have been countered by other operators. In both applications of the ground-patrol operator, army.1 was instantiated to be the particular meachnized unit, mechb, because this unit met the constraints on firepower, mobility, location, etc. The
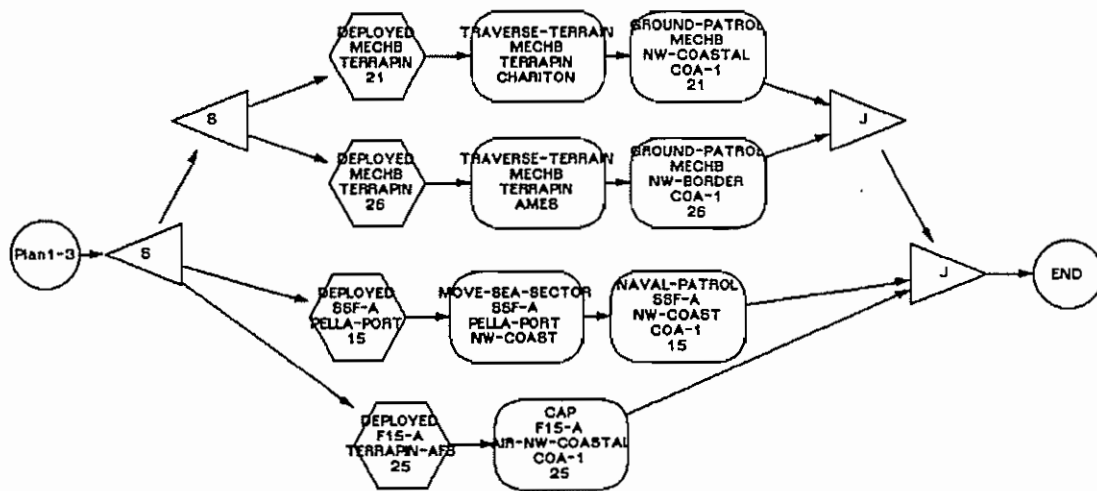
18

Figure 8: Plan after Deterring Threats (Level 3)

plan requires the mechb to do ground patrols simultaneously in the NW-Coastal region and the NW-Border region (because region.1 was instantiated differently in the two operator applications). The planner ensures there are no conflicts in the plan and then plans another level by a similar application of operators to the newly produced plan. The next planning level will use deployment operators to solve the deployment subgoals that are now in the plan.

4.2   The User Interface in SOCAP

The Socap interface provides facilities for guiding the user through the plan generation process. The amount of user interaction can be varied during the planning process. It can range from being fully automated, in which case a plan is generated with no human interaction; to semiautomated, in which the user makes some choices; to fully manual, where the user makes all the choices. At each goal in the plan, the user can request the relevant operators that achieve the goal to be displayed. Likewise, when attempting to bind a variable associated with an argument of an operator, the possible bindings can be displayed. For instance, the user may be presented with the set of military units that have the appropriate capabilities to deter an enemy threat, or a list of suitable locations for the military operation. This set may be constrained by the

19

preconditions and other constraints associated with the arguments of the relevant Sipe operator. At the end of each plan level, the plan is checked for logical consistency, and then progresses to the next level until there are no more goals to be satisfied or actions to be decomposed further.

The plan may be displayed at each plan level, either as a partially ordered network of actions and goals, or on a time-based map display. The map display shows the actions that are occurring on different days during the mission. The temporal information for the map display is derived from durations associated with each action and from the dates when the enemy threats should be deterred or countered. Clicking the mouse on a token on the map display will bring up a window describing the military operation represented by the token, and the location, times, and units of the operation. A second mouse click can be used to bring up another window with further information about a specified military unit, e.g., its current military status or mobilitization status.

Section 5 describes more of the techniques used to encode information. The plans generated so far have 100 to 200 executable actions in the final plan. (The entire plan has many times this number of nodes, since some nodes are conditions which help record the rationale behind the plan and information for repairing it.) The major forces identified in the plan are then further decomposed by FMERG to their TPFDD-level components, resulting in 2,500 entries in the TPFDD. For a large-scale military operation, such as in the Gulf crisis, the number of TPFDD-level entries would probably be an order of magnitude larger. The IFD-2 knowledge base is by no means complete. It has fleshed out the skeleton of a much larger set of plan operators, including predicates for describing the current situation, and classes and objects for capturing static information. A rough estimate for an operational Socap knowledge base is given in Table 1.

5   Lessons Learned

The lessons learned in applying SIPE–2 to military planning can be divided into three sections: successes in applying the existing technology, difficulties, and future research.

Data structure	IFD-2	Operational
classes/objects	500	2,500
properties per object	5 to 15	50 to 100
predicates	1,500	15,000
operators	50 to 70	500 to 1,000

Table 1: Size of Knowledge Base for Current SOCAP and Operational SOCAP

5.1 Successes

The Sipe hierarchical planning process maps well to the military operations planning process. As a result, it was relatively easy to group sets of operators according to the various phases/levels of the operations planning process, described earlier. The hierarchical structure of the operators also encourages a similar structure for the predicates that describe the world. Hence, the preconditions of operators are excellent means of bringing situational data and information to bear at the relevant planning level and at the appropriate level of detail. For instance, general locations of enemy threats and high-level geographic features may be appropriate for the identify-threats level of the planning process, but more detailed threat assessment, e.g., force composition, strength, and terrain analysis, are required for the employment planning phase. Furthermore, the hierarchical structure maps well to the chain of command, described in Section 2.

The sort hierarchy provides a clear representation of static information within SOCAP. This static information is captured in the properties of classes and objects that are often used in posting constraints on the arguments of operators. For instance, the choice of an airfield for a deployment action may rely on having the property runway-length be long enough for available air transports, such as C-5s, C-141s, and C-130s. There is a clear distinction between static information, represented in the sort hierarchy and as predicate statements, and dynamic information, represented only by predicate statements. This distinction is important for two reasons: (1) the retrieval

21

of static information is more efficient than the retrieval of dynamic information, since the system does not need to reason about the effects of actions, and (2) the validation of static information is helped by the tree structure of the sort hierarchy.

The arguments of operators and preconditions are represented as planning variables with constraints. This provides a means of delaying the binding of specific instances to these variables until the constraints select the most appropriate instance. This least-commitment approach to variable binding is an important capability provided by SIPE-2, that is heavily used in SOCAP. SOCAP also makes use of the planner's ability to force the binding of these variables with user guidance. For instance, this facility may be used to force the selection of favored military units for specific operations.

SIPE-2 provides a mechanism for permitting situational knowledge to determine the number of specific subgoals introduced into a plan. For instance, in order to determine how many enemy threats to counter, the system checks the number of enemy threat units identified in the threat assessment database and generates a subgoal for each. This mechanism permits iteration where the number of iterations is determined by the situational data, rather than provided in advance. There are a variety of these iterative operators that search for different types of enemy threats, whether ground, sea, or air. Section 5.2 discusses problems concerning the number and choice of subgoals introduced by this mechanism.

SIPE-2 permits a great deal of information to be presented to the user at a variety of levels of detail. As a result, the GUI has many menus that can be called to access this information, and using it effectively requires some knowledge of the workings of the planner. The GUI is thus better suited for developing applications than for using an application operationally. The Socap interface, on the other hand, is tailored to the military crisis action planning environment. In particular, it stresses the interactive nature of the planning process, and extracts appropriate details during the planning process and presents them to the user. Thus, when a user is viewing the possible choices of military units for an operation, SOCAP presents the constraints that led to these choices. It is possible to view all the possible instantiations of a planning variable, given the constraints associated with the variable, whether it represents a

military unit, resource, location, or time. Operator choices for extending the plan are displayed, and the plan structure produced is highlighted, including the effects achieved and the preconditions that must be satisfied. The ability of the Sipe GUI to highlight specific information about the plan or its structure is particularly useful in the Socap interface. This includes highlighting nodes that contain certain predicates or arguments, and highlighting the predecessors, successors, and actions in parallel for a given action.

A time-based map display provides another means of displaying the plan that is particularly appealing to military planners. This map display is currently being developed to display situation data or overlays on a digitized map. It is possible to show the operations that occur on each day of the mission and display appropriate information about the type of military operation, the units involved, and the boundary of the operation. In this way, the user can query what is going on, where and when, in the context of map information. We have used this map display to represent both the employment plan on a detailed map of the area of interest and the deployment plan on a coarser-grained map.

5.2   Difficulties

SOCAP did not extensively use the Sipe resource reasoning capability for three reasons. First, dealing effectively with resource conflicts involves balancing various tradeoffs. This requires complex representations of hard and soft constraints and mechanisms for constraint relaxation. Most AI planning systems (including SIPE-2) do not have these capabilities, and instead rely on the representation of hard constraints and constraint satisfaction algorithms. Second, an algorithm tailored to this particular domain would be necessary to intelligently assign resources. For instance, when choosing military units for operations, in order to minimize the number of troops involved in the operation, it is often wise to choose units already involved in the plan, except when they have been overused. In addition, there must be information to allow balancing such a desire with other conflicting heuristics.

23

Finally, although SIPE-2 does have a mechanism for representing shareable resources, it must be determined in advance how such resources might be shared. This is too inflexible for military operations planning. For instance, a large military unit, such as a division, may be employed in several operations simultaneously, where each operation uses some of the division's capabilities. The number of operations over which the division may be shared depends on the amount of resources required for each operation. Thus, the only way to reason about the shared resource is to consider the capabilities of the division as a consumable resource purely for this specific set of operations.

Each of these three problems is difficult and best addressed with technology that relaxes soft constraints in search of an optimal solution, e.g., the scheduling technology at CMU. An integration of planning and scheduling, as described in Section 6, is needed to address these problems.

Another difficulty is the limited temporal reasoning in SIPE-2. Two Sipe actions are either ordered with respect to each other, or they are unordered. If the latter, the planner considers it possible to order them in either order or execute them simultaneously. Information about start times and durations is only used when it can be deduced that two actions should be ordered based on this information. This is limiting; for example, one cannot model when during the execution of an action its various effects become true, nor can one model actions that *must* occur simulatenously. Allen's 13 temporal relations [1] would have been useful. This would have permitted more versatile operations with explicit representation of actions starting or finishing at the same time, overlapping each other, or one occurring during another. Many dependencies between different military actions could have been represented in this way. For instance, cargo offload teams should arrive at the same time as the first air or sea transport arrives at the air or sea port, the deployment of troops to their destination should overlap the deployment of supplies and equipment, and close air support should take place during a ground offensive operation.

Another useful capability is the serendipitous combination of subgoals. For instance, at present, for every enemy threat identified, a friendly unit is identified to deter or counter it. If several small enemy forces are located close to each other, SOCAP

24

attempts to deal with each threat individually, rather than considering them as an aggregate threat that might be countered with a single, larger, friendly force. Whether the aggregation is done by the user or by some algorithm, it is important that the original subgoals be replaced by a new subgoal.

Currently, it is cumbersome to represent the notion of a task force whose composition is determined dynamically by whichever military units were assigned to lower-level actions. It is possible to represent a class of objects of type task-force and make use of a part-of predicate to relate specific military units to a specific task force, but this is not perspicuous. This notion of a higher-level term used to describe the combination of lower-level objects is an important aspect of military planning. For instance, a military brigade that comprises three battalions may be split into two or more battalion task forces that could make use of the battalions within the brigade, or additional units and resources from other brigades or divisions. It is important to make reference to the battalion task force at higher levels of the plan, especially when coordinating with other task forces or higher-level units. It is also necessary to ensure that the capabilities of the task force are explicitly represented so that they are not overutilized.

Another problem involves the lack of an explicit mechanism for reasoning about the order in which goals should be achieved. SIPE–2 supports operators that delay solving of a goal for one planning level when the preconditions of the operator are true. For instance, one may decide to achieve all employment goals first and only start on the deployment goals when the employment goals have been satisfied. However, having to encapsulate such a heuristic in the preconditions of an operator is not ideal. One solution is to permit the developer to specify an algorithm to weigh trade-offs between several heuristics that determine which goals to satisfy next. This would allow expansion of some parts of the plan down to a detailed level, while other parts of the plan might be kept at a higher planning level, thus allowing details associated with one part of the plan to be specified before another part of the plan, which depends on these details, is formulated. The O-Plan architecture provides a framework for doing this type of reasoning [3].

6 Future Work

This section describes the important issues that SRI plans to address in the future, in conjunction with other research groups in the Planning Initiative. During the planning process, many decisions have to be made concerning the choice of operations, military forces, resources, locations and times. To reason about the various possibilities requires using a variety of simulators, case-based reasoners, and decision-theoretic and uncertain reasoning aids for guiding such choices. The simulators may perform feasibility analyses for combat, logistics, transportation, command and control and the like. Case-based reasoners would explore the similarity between the current situation and previous examples (or cases) of missions that employed specific operations, forces, and resources, at certain locations and times. Decision-theoretic aids and uncertain reasoning techniques can be applied to determine the utility of specific choices, and to provide probabilistic measures of success that could be used to determine the alternatives to be explored and enumerated. The above techniques could be applied during the planning process to guide choices, or afterward to evaluate the plan(s) developed. Such techniques could also provide metrics for identifying qualitatively different COAs.

During the Socap project, a simple simulator was used as a postprocess to determine the transportation feasibility of the COA. The output of the simulator included a list of the number of military forces that were predicted to arrive late at their destinations. The input to this simulator required specific times to be associated with the movement actions in the deployment plan. Ideally, a TPFDD is the proper input to the simulator. The Planning Initiative includes work on simulation and scheduling systems that could be used for determining transportation feasibility.

6.1 Replanning and Knowledge Acquisition

During IFD-2, the emphasis was on demonstrating the feasibility of applying plan generation techniques to the crisis action planning process. By stressing the generation of plans, it was possible to focus attention on describing the different types of military

operations, forces and their capabilities, and using situational data to guide the necessary choices during planning. However, in a crisis the situation is changing rapidly. Previous decisions need to be revised and new goals need to be satisfied. Replanning techniques are required to reuse as much as possible of the existing plan while modifying it appropriately.

To complicate matters further, not only is the situation changing rapidly, the environment is uncertain. It will be necessary to develop techniques to reason about how the uncertainty in the situational data affects the planning choices. This is important both for selecting operators, which may now have uncertain effects, and for selecting units for use in the plan. Exploring the use of uncertainty measures to determine the robustness of plans or the sensitivity of plans to changes is also important. An ongoing effort at SRI is investigating planning and executing in uncertain environments.

Military operations planners who viewed SOCAP often asked about support facilities for updating and writing new operators. This involves providing facilities for checking that the preconditions and effects are syntactically and semantically correct. An ongoing effort at SRI [7] has specified the ACT formalism and developed an interactive editor for inputing operators in this formalism which can be translated to SIPE-2. However, algorithms to ensure that the revised or new operators do not adversely affect other existing operators would also be needed. This is a difficult problem that may provide an excellent domain for machine learning techniques.

6.2   Integrating Planning and Scheduling

Another important research area is the relationship between, and integration of, planning and scheduling techniques. Current research work (involving SRI and CMU) is investigating the integration of SIPE-2 with the OPIS (Opportunistic Intelligent Scheduler) scheduling technology developed at CMU. This research will be conducted in several phases.

The first phase will provide a Sipe plan as input to the OPIS system, and send feedback to the planner concerning resource conflicts that present difficulties to OPIS.

The second phase will examine the integration of capacity planning techniques during the plan generation and replanning processes to aid the early assignment of resources based on projections of predicted resource bottlenecks. The third stage will investigate more closely the guidance to the scheduling process that can be provided by the dependency structure of the plan, the choices made during the plan generation process, and, the alternative choices that are recorded in the plan state.

We will examine how information from the scheduling process might be used to guide replanning. For example, the scheduler might provide information that suggests which alternative miltary actions should not be considered because they make use of over-utilized resources. Providing such information during the replanning process would reduce resource conflicts during the subsequent re-scheduling process. Controlling the planning and scheduling processes poses some difficult questions, such as: When should the system stop plan generation and run the scheduler or can they run simultaneously? When can the scheduler repair a plan and when must the planner be used for plan repair? When should the system simulate the execution of the plan and how should the simulation interact with the planner and scheduler?

7   Conclusions

SOCAP successfully demonstrated that AI planning techniques can be used for the generation of large-scale military OPLANs. It provides the first steps toward an operational prototype that can be tested on real military crises, although it has so far been tested on only one scenario. It shows how the domain requirements of the application drive research in AI planning technologies, and how rapidly these evolving tools and techniques can be transferred to the application.

SIPE-2 supported efficient plan generation for this scenario. The effectiveness and efficiency stemmed from its hierarchical planning process (which naturally fit the hierarchical structure of the domain), extensive use of its sort hierarchy for encoding military databases, the ability to place constraints on planning variables, its powerful graphical interface for viewing plans and data, and the interactive planning capabil-

28

ities. Several difficulties were identified, however, including the inability to reason about complex temporal information, lack of an ability for encoding a domain-specific algorithm for assigning resources to actions, an inability to combine subgoals, and lack of a mechanism for reasoning about when goals should be achieved.

Producing decision aids that will help military planners in an operational setting will involve the integration of several ongoing research areas in the Planning Initiative. These include machine learning, scheduling, plan repair, case-base reasoning, simulation, and uncertain reasoning techniques.

Acknowledgements

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the Association for Computing Machinery*, 26:832–843, 1982.

[2] M.A. Bienkowski. Initial Requirements Analysis for Automation in Support of Joint Military Planning. Technical Report ITAD-2062-TR-92-40, SRI International Artificial Intelligence Center, Menlo Park, CA, March 1992.

[3] K. Currie and A. Tate. O-plan: The open planning architecture. *Artificial Intelligence*, 52(1):49–86, 1991.

[4] S.F. Smith. A constraint-based framework for reactive management of factory schedules. In M. Oliff, editor, *Proceedings of the International Conference on Expert Systems and the Leading Edge in Production Planning and Control*. Charleston, SC, May 1987.

[5] David E. Wilkins. *Practical Planning: Extending the Classical AI Planning Paradigm*. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1988.

[6] David E. Wilkins. Can AI planners solve practical problems? *Computational Intelligence*, 6(4):232–246, 1990.

[7] David E. Wilkins. Planning in dynamic and uncertain environments. Annual report, SRI International Artificial Intelligence Center, Menlo Park, CA, September 1992.

[8] David E. Wilkins. *Using the SIPE Planning System: A Manual.* SRI International Artificial Intelligence Center, Menlo Park, CA, 1992.